코퍼스 확률에 기반한 한국어 표준발음 생성

김동성*
한국외국어대학교/언어인지학과/BK21사업팀
dsk202@hufs.ac.kr

The Corpus-probability Based Generation of Korean Standard Pronunciation

Kim, Dong-Sung
Dept. of Linguistics & Cognitive Science/Hankuk U of Foreign Studies

요 약

본 연구에서는 코퍼스 확률에 기반하여 한국어 표준 발음 생성에 대한 연구를 한다. 기존의 이은영 외 (2005)에서 연구된 규칙기반의 한국어 IPA 발음 변환방식과는 달리 본 연구에서는 음운변환 코퍼스를 바탕으로 표준발음을 변환한다. 이 방식을 위해서 Brill(1995)에서 제안한 변형기반 학습방식이 활용되었으며, 단계적인 처리방식이 아닌 입-출력 대응 방식의 확률적 처리 방식이 제안되었다. 음운변환 방식은 음운규칙에 근거한 처리가 아닌 언어자원인 코퍼스를 활용해서 처리하였다는 점에서 기존의 연구방식과 차이가 있다. 또한, 기존 연구에서는 음운규칙을 단계적으로 적용하여서 입력형이 출력형으로 도출되기 위해서 여러 단계를 거쳤지만, 본 연구에서는 입력형과 출력형의 일대일 대응이라는 점에서 차이점을 보인다.

1. 머리말

본 연구는 음운변환 코퍼스를 활용해서 표준국어 발음을 생성하고자 한다. 이러한 방식을 위해서 본 연구에서 활용한 것은 Brill(1995)에서 제시한 변형기반의 학습방식이다. 또한 여러 단계의 음운변환의 단계를 거치지 않고, 코퍼스의 확률을 활용해서 입력형과 출력형을 일대일로 변환하는 작업을 하였다.

음운변환 작업의 단계를 간략하게 설명하면 다음과 같다. 먼저, 음운변환 코퍼스를 위해서 세종코퍼스의 구어 코퍼스를 대상으로 이은영 외 (2005)에서 제시한 '한국어 표준발음 IPA 변환기(이하[표준발음변환기])'를 활용해서 음운변환 코퍼스를 작성하였다. 이 코퍼스는 입력형과 출력형이 서로 대응되어 있는 병렬형태의 음운 코퍼스이다. 이를 토대로, 입력 음운과 출력 음운 단위로 대응관계를 추출하고, 확률 정보를 추출하였다. 이러한 정보를 변형기반의 학습방식을 활용한 음운변환에 활용하였다. 이 연구는 음운론적 언어처리에서 확률기반을 적용하는 것은 규칙기반이 아니라 코퍼스를 활용한 데이터 기반이라는 점에서 그 의의를 지닌다.

* 이 연구에 참여한 연구자는 '2단계 BK21 사업'의 지원비를 받았음.

규칙기반의 언어처리는 Gildea and Jurafsky(1996), Daelemans et al.(1994), Kaplan(1994), Karttunen(1993; 1998)에서 연구되었던 방식으로 음운규칙의 단계별 적용을 원칙으로 한다. 이 연구에서는 이러한 방식을 거치지 않고, 입력형-출력형 대응방식을 활용하였다.

본 연구에서는 음운변환 코퍼스에서 발견되는 확률정보를 활용해서 이를 적용하였다. 이러한 방식은 기존의 처리 방식과는 다르게 코퍼스라는 언어자원을 활용하고, 입력형과 출력형만을 대응하기 위한 노력과 일치한다. 확률적 기반의 처리는 여러 단계에서 얻어야 할 형태 및 음운의 문맥적 정보를 제공하기 때문에 단계별 음운 규칙 적용이 없이 음운변환 작업을 코퍼스 기반에서 가능하게 하였다. 실제 본 연구의 작업은 기존의 연구방식과 다르게, 음운 변환에서 코퍼스 기반의 적용이 가능함을 보여주었다. 그리고, 향후 연구방식에서도 음운 데이터를 활용하는 변환방식이 효과적임을 보여 주었다.

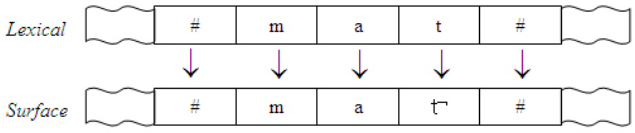
본 연구의 구성은 다음과 같다. 2절에서는 기존 연구를 설명한다. 3절에서는 본 연구에서 활용된 변형기반 학습방식을 적용한 확률적 음운변환을 상세히 설명할 것이다. 4절에서는 음운변환의 과정과 실험에 대한 부분을 설명할 것이다. 5절은 이 논문의 결론이다.

2. 관련 연구

규칙기반의 음운처리 방식은 Chomsky and Halle(1968)와 일치하는데, 다시쓰기 규칙(rewriting rule)에 의해서 입력형이 출력형으로 제한된 문맥환경에서 변화한다. 예를 들어서, 한국어의 경우에 일반적으로 폐쇄음(stops)이 불파 무성 폐쇄음(unreleased voiceless stops)으로 어말위치에서 변하게 된다. 이러한 음운변환을 규칙으로 만들어보면 (1)과 같다.

(1) t → t˺ /_ #

이러한 변환 규칙은 여러 방식으로 연산될 수 있으나, Koskienniemi(1984)는 입력형과 출력형을 일대일로 대응하는 이단계형태론(Two-level Morphology)'을 제안하였다. 이러한 방식은 [그림1]과 같이 표현될 수 있다.



[그림 1] 이단계형태론

'이단계형태론'은 입력형을 정해진 문맥환경에서 출력형을 대응하는 방식으로 음운변환을 연산한다. 그러나, 음운규칙은 입력형의 음운환경에 따라서 매우 다르게 적용될 수 있으며, 이러한 음운환경의 다양한 환경을 '이단계형태론'이 적절하게 소화하지 못한다. 예를 들어서, 한국어의 경우에 입력형이 활용에 의한 표면형이면 구개음화(palatalization)를 겪게 되지만, 기타의 형태라면 구개음화를 겪지 않는다. 대표적인 예로, '잔디'와 '굳이'가 각각 [잔디]와 [구지]로 다르게 발음이 된다. '굳이'는 활용에 의한 것으로 구개음화를 겪게 되지만, '잔디'는 이 경우에 해당하지 않으므로 구개음화를 겪지 않는다. 이와 같이 음운규칙이 입력형에 따라서 다르게 적용이 되므로, '이단계형태론'이 이런 측면을 고려하게 되면 매우 복잡하거나 처리가 어렵다.

이러한 '이단계형태론'의 문제점에 대해서 전산음운론(computational phonology)에서는 전통적인 SPE 식의 '규칙순서화(rule ordering)'를 활용하는 음운처리가 많이 활용되었다(Gildea and Jurafsky 1996; Daelemans et al. 1994; Kaplan 1994; Karttunen 1993, 1998).

이러한 '규칙순서화'는 규칙의 양과 순서에 따라서 규칙의 순서가 기하급수적으로 많아 질 수 있으며, 중간단계에서 약간의 순서가 뒤바뀌어도 출력형이 매우 다르게 된다. 따라서, 정확한 출력의 문제와 '규칙순서화'를 단순하게 전산적으로 단순하게 만들고자 하는 많은 연구가 있었다(Gildea and Jurafsky 1996; Daelemans et al. 1994).

반면에, '이단계형태론'은 음운처리보다는 형태론의 처리에 많은 비중을 두면서, 일대일 대응방식을 다루었다. 이러한 점은 Brill(1995)에서 제안한 변형기반 학습방식과는 차이가 있지만, 크게 일대일 대응방식에서 일맥상통하는 점이 있다. Brill(1995)는 일련의 형태 태깅의 문제를 다루면서, 좌우문맥을 기계적인 학습방식을 제안하였다. 이 방식은 일련의 변형규칙을 템플릿(template)에 산정해 놓고 입력형을 변형하는 것이다.

음운변환의 경우에 변형기반의 학습방식을 적용한 연구는 현재 발견되지 않으며, 매우 적은 양의 코퍼스 기반의 연구가 발견된다. Daelemans et al.(1994)는 코퍼스를 활용하여 통계적 음운처리기법을 연구하였다.

이 연구는 코퍼스를 활용하여 통계적인 기법을 연구하고자 한다. 또한 음운 코퍼스를 학습하는 방식을 제안하는데, Brill(1992)에서 제시한 변형기반 학습방식을 활용한다. 코퍼스에서 발견되는 규칙을 학습하는데 확률적 통계처리방식을 활용하였다.

3. 변형기반 학습, 확률적용과 음운변환

변형기반 학습은 정해진 코퍼스를 통해서 일련의 규칙을 추출하고 이를 입력형에 대해서 출력형으로 변환하게 된다. 한국어에서 무성 치경 폐쇄음(labio-dental voiceless stop)인 /t/는 음운환경에 따라서 [t, t˺, d]로 변환한다.

(2) 템플릿의 예
① 이전 음운환경이 #_ (어절 처음) 이면, t는 t로 변한다.
② 이전 음운환경이 모음이면, t는 d로 변한다.
③ 이전 음운환경이 자음이면, t는 t로 변한다.
④ 이전 음운환경이 _# (어절 말) 이면, t는 t˺로 변한다.

템플릿은 코퍼스에서 발견되는 모든 환경을 조사하여서, 음운변환 규칙을 만들어 진다. 하나의 입력형이 입력되면, 관련된 환경의 음운규칙을 찾아서 하나씩 변형하게 된다. 모든 변형규칙이 적용되면, 입력형 변환이 종료되고 최종 출력형으로 출력된다.

이 방식은 '이단계형태론'과 유사해 보이나, 실제적으로 데이터 학습에 따른 처리방식이라는 점에서 차이가 있다. 데이터의 규칙을 학습함으로써 규칙변형이 이루어지게 되며, 더 이상 적용할 변형규칙이 없을때까지 음운변환을 진행한다.

변형기반 음운변환을 위해서 코퍼스를 학습하기 위해서는 음운변환의 환경을 학습하여야 한다. 이러한 음운변환 환경은 문맥 윈도우(context window)개념을 도입하여서, 처리하였다. 문맥 윈도우는 코퍼스 언어학의 '문맥내 색인어(KeyWord In Context)' 기법과 동일한데, 하나의 중심 윈도우의 색인어를 중심으로 몇 개의 윈도우가 좌우로 떨어져 있는가가 다루게 된다. 9개의 윈도우를 다룬다면, 중심어를 중심으로 좌우로 모두 9개의

음운환경을 포함한 것이다. 아래의 그림은 문맥 윈도우가 9, 7, 5개인 윈도우이다.

[그림 2] 문맥 윈도우와 음운환경

템플릿 예인 (2)에서는 음운규칙 변형규칙의 환경이 2개의 문맥윈도우인 것을 보여준 것인데, [그림 2]는 음운환경이 9, 7, 5개의 문맥윈도우로 확장된 것을 보여준다. 실험 실제 환경에서는 20, 10, 5, 4, 3, 2개의 문맥윈도우가 활용되었다.

코퍼스에서 발견되는 변환규칙은 (2)에서와 같이 코퍼스에서 규칙을 추출하였다. 이러한 음운변환의 규칙은 코퍼스에서 실제로 발견되는 것로 구성되므로, 규칙이 다른 코퍼스에서도 발견되는지에 대한 통계적 신뢰성 검증이 필요하다.[1] 이러한 문제점에 대해서 실제로 발견되는 코퍼스의 규칙의 신뢰성을 측정하였다. 이를 위해서 코퍼스에서 발견되는 규칙에 대해서 전체 규칙 적용의 범위를 확률 연산하였다. 만약 'ㅌ→ㅊ /_#' 와 같은 규칙이 있다면, 환경의 범위인 어절 말 환경에서 /ㅌ/가 있는 경우를 모두 산출하고 이 중에서 /ㅌ/가 [ㅊ]로 변화한 경우를 모두 산출하였다. 이를 통해서 전체 규칙 적용의 범위에서 규칙이 적용된 경우를 확률적으로 구하게 된다. 하나의 규칙 r을 확률 $P(r)$ 로 구하는 수식은 다음과 같다.[2]

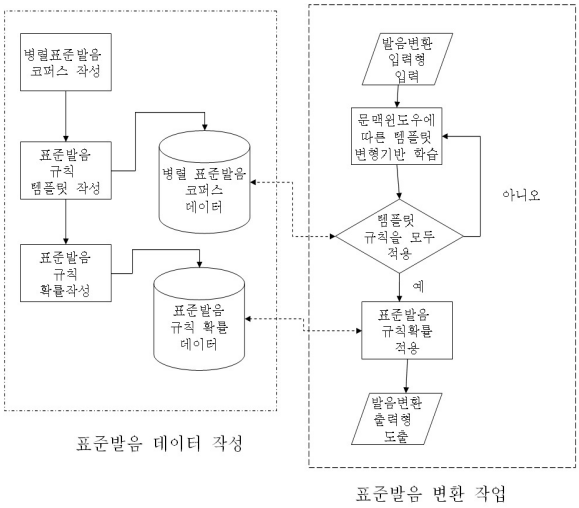
(3)

이러한 확률적 방식은 수의적 발음의 경우에 음운결정의 문제를 해결한다. 실제 코퍼스에서 발견되는 음운변환 중 음운환경이 동일하지만 음운변환이 다른 경우도 발견이 된다. [표준국어발음]은 수의적 발음을 인정하고 있다. 가령 '신문'의 경우에는 [신문] 또는 [심문]으로 발음할 수 있다. 따라서 어떠한 발음을 할지에 대한 결정의 문제가 있다. 이러한 경우에 동일한 음운환경에서 발견되는 문제에서 동일한 음운규칙을 동시에 수의적 적용하지 않고, 확률이 높은 규칙을 적용하였다.

4. 음운변환과 실험

4.1. 실험절차

전체 실험의 절차를 나타내면 아래 그림과 같다.



[그림 3] 전체 실험진행

실험에 활용된 것은 21세기 세종계획 2단계(1998년에서 2002년까지) 중 세종코퍼스에서 구어 코퍼스를 활용하였다. 대략적으로 14,500어절의 코퍼스가[3] 구어 코퍼스로[4] 강연이나 기타 구어체를 발음 나는 대로가 아닌 [표준맞춤법]에 맞게 전사한 것이다. 따라서, 표준맞춤법에 맞게 전사되었으므로, 실제 표준발음과는 다르다. 국어의 [표준국어발음]에 맞게 전사하는 것은 현대 한국어 발음규칙에 맞게 전사하는 것을 말한다. 예를 들어서, /손가락/의 경우에는 [손까락]이나 [송까락]으로 전사되어야 한다.

자료에서 활용된 음운기호는 IPA(International Phonetic Alphabet)기호가 아닌 IPA를 텍스트 기호로 전환한 ARPabet이다.[5] 한국어의 음소를 19개의 자음과 10개의 모음을 다음과 같이 표현하였다.

1) 이러한 점은 통계적 코퍼스 언어학의 평탄화(smoothing)의 문제점과 일치한다. 코퍼스의 양과 질에 따라서 다른 통계적 수치가 도출될 수 있으며, 이러한 문제점은 전체 통계집단을 다시 고려해야 하는 문제점을 야기한다. 자세한 점은 Jurafsky and Martin(2000) 참조. 본 연구에서는 평탄화를 직접적으로 적용하지 않고 일련의 확률을 최대우도추정치(Maximum Likelihood Estimation)로만 산출하였다.

2) 비슷한 연구에도 (3)과 유사한 확률적 연산이 제시되었다. Hong and Huang(1998)와 Albright(출간예정) 참조.

3) 대략적으로 60,000개의 형태소를 가지고 있고, 1어절당 대략적으로 4.2개 정도의 형태소를 갖는다. 코퍼스는 106,478개의 음소로 구성되었고, 한 형태소당 1.78개의 음소가 활용되었다.

4) 구어 코퍼스를 활용한 것은 발음변환 작업을 위해서 구어로 코퍼스를 통해서 실제 한국어 표준발음의 발음을 추적하기 위해서이다.

5) ARPabet을 선정한 이유는 IPA 코드는 Unicode나 ASCII코드가 아니므로 실제 다른 텍스트 처리에 있어서 문제가 있다. 따라서, 처리의 용이성을 위해서 ASCII 코드로 된 음소기호를 채택하였다. http://en.wikipedia.org/wiki/Arpabet 참조

IPA	ARPabet	IPA	ARPabet
p	p	i	i
pʰ	ph	ɛ	E
p*	p*	ɯ	W
t	t	ʌ	A
tʰ	th	u	u
t*	t*	o	o
k	k	ɑ	a
kʰ	kh	ɯ	Y
k*	k*	j	y
s	s	w	w
s*	s*		
tɕ	ts		
tɕʰ	tsh		
tɕ*	ts*		
m	m		
n	n		
ŋ	G		
l	l		
h	H		

[그림 4] 한국어 ARPabet

표준발음 코퍼스를 활용하기 위해서 이은영 외(2005)에서 작성한 [표준발음변환기]를 활용하였다. 이 [표준발음변환기]는 텍스트로 입력된 철자를 표준발음에 맞게 전환하는 역할을 한다.[6] 구어 코퍼스를 [표준발음변환기]를 활용해서 한국어 표준발음으로 변환하고, 한국어 맞춤법 표준안으로 전사된 것과 표준발음으로 변환된 것을 같이 정렬하였다[7]. (4)는 병렬 표준발음의 예이다.

(4) 나는 집에 간다.
나는 {n.a-n.W.n}/{n.a-n.W.n}
집에 {ts.i-p.e}/{ts.i-b.e}
간다 {k.a.n-t.a}/{k.a.n-d.a}

위에서 '/' 를 경계로 왼쪽은 구어 코퍼스, 오른쪽은 표준발음이 정렬되었다. '.' 는 음절 내 구조인 초성(onset), 중성(rhyme), 종성(coda)을 구분한다. 그리고, '-' 는 음절 경계를 표시한다. (4)에서 명기된 방식은 입력형과 출력형을 대응한 것이다. 입력형은 한국어 맞춤법표준안으로 표기된 음소기호이고, 출력형은 한국어 표준발음기로 변환한 음소기호이다. 각각의 발음열을 정렬하여서 실제 발음변환 및 음절 구조의 변화를 대응하였다.

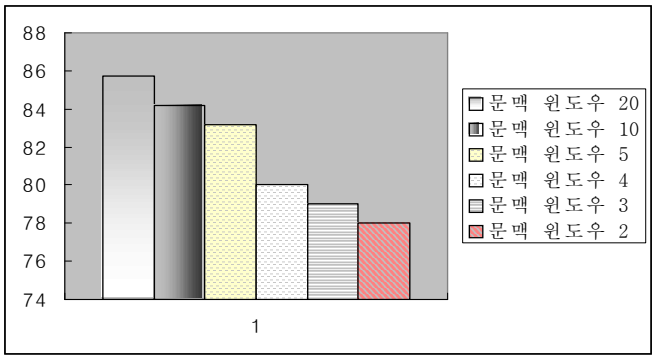
음운규칙은 문맥윈도우를 정렬하여 추출된다. 문맥 윈도우는 발음열에서 발견되는 음소들의 크기를 말한다. 만약 윈도우가 4개로 되고, /t/의 음운이 [t] 또는 [d]로 변환되는 규칙을 추출한다면, (5)와 같이 추출될 것이다.

(5)
① {#,#,_,$\text{Vowel}_{\text{Rhyme}}$}의 환경에서 t는 t로 변한다.
② {$\text{Vowel}_{\text{Rhyme}}$,-(음절경계),_,$\text{Vowel}_{\text{Rhyme}}$}의 환경에서 t는 d로 변한다.

이러한 규칙과 더불어 규칙적용의 확률을 구하여, 데이터를 작성하였다. 이와 같은 데이터를 바탕으로 음운변환의 입력형이 입력되면, 템플릿의 데이터를 변환하게 된다. 변환하는 템플릿 데이터는 문맥윈도우가 각각 20, 10, 5, 4, 3, 2개인 것으로 구분되었다.

4.2. 실험결과

실험을 위해서 1,000여개의 어절을 세종코퍼스에서 무작위로 추출하여서, 발음변환을 하고 실제 표준발음과 비교하였다. 문맥 윈도우는 20, 10, 5, 4, 3, 2개로 변환하면서 정확도의 변화를 살펴보았다.

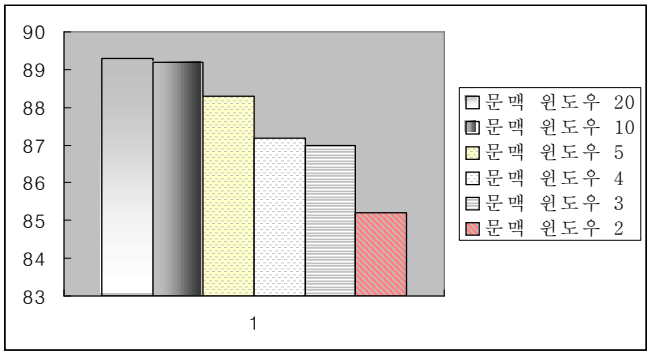


[그림 5] 문맥 윈도우 크기와 정확도 1

위의 [그림 5]를 통해서 알 수 있듯이, 윈도우의 크기가 크면 클수록 전체 정확도가 늘어난다. 윈도우의 크기가 크다는 것은 음운환경의 정보가 많다는 것을 의미한다. 따라서, 실제 음운변화를 위해서는 음운정보의 양이 많아야 한다는 것을 의미한다. 코퍼스를 조사하면, 하나의 어절은 7음절 정도로 구성되며, 하나의 음절은 대략적으로 1.78 음소로 구성된다. 음운환경의 정보가 5이상일 경우에 정확도가 많이 향상되는 것을 감안하면, 실제 음운환경이 3 음절 이상의 정보가 필요하다. 이것은 음절 내부의 정보도 중요하지만, 음절 경계를 벗어난 정보가 많이 필요한 것을 말해준다. 또한 음절 보다 더 큰 음운구조의 정보가 필요하다. 신지영, 차재은(2004)에서 한국어 음운변환이 이루어지는 음운구조의 단위를 운율 및 강세구 단위까지 고려하는 것을 감안하면, 음운정보의 양이 많아지면 정확한 음운변화가 가능한 것을 설명할 수 있다.

확률적 처리를 위해서 전체 음운변화가 가능한 음운환경에서 실제 규칙이 적용된 범위를 연산하는 (3)의 방식을 적용하였다. 이 방식도 문맥 윈도우의 크기를 조절하면서 정확도를 측정하였다.

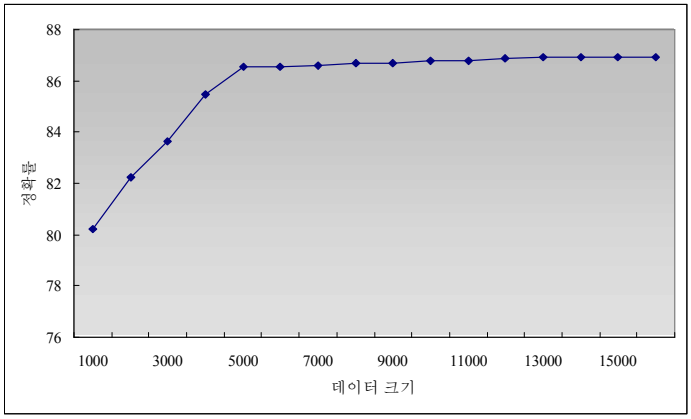
6) http://urimal.cs.pusan.ac.kr 참조
7) 이은영 외(2005)의 표준발음 변환기는 98%의 정확도를 갖는다고 한다.



[그림 6] 문맥 윈도우 크기와 정확도 2

측정결과 확률적 고려를 한 경우가 윈도우 크기를 더 작게 한 경우에도 정확도의 향상이 많았다. 이것은 확률적 정보가 더 정확한 음운변환을 고려하게 하여서 더 적은 음운환경에 대한 정보로도 더 정확한 예측을 가능하게 한 것으로 해석된다. 확률적 정보는 하나의 음운환경에서 예측되는 음운규칙을 더 정확하게 예측하므로, 확률정보를 적용하는 것은 정확도의 상승을 가져오게 된다.

또한, Brill(1995)가 살펴본 바와 같이 실제 변형기반 학습방식은 실제 학습 코퍼스의 크기가 일정비율 이상이 되면 줄어든다는 점을 확인하였다. 학습 코퍼스의 크기를 1,000여개 이상을 늘리면서 실제 코퍼스의 크기와 정확도 사이의 상관관계를 살펴보았다.



[그림 7] 데이터 크기와 정확률의 변화

[그림 7]과 같이 데이터의 크기가 5,000여 개 정도가 되었을 때, 정확도의 변화는 정점으로 가고 있다. 데이터의 양이 5,000여 개보다 더 많아 져도 정확도의 큰 변화가 없는 것을 알 수 있다. 이러한 점은 데이터의 양이 많아지면 질수록 정확도의 증가가 발생하는 것이 아니고, 일정한 양의 데이터가 실제 학습에 필요함을 보여준다. 이것은 Brill(1992)에서 관찰된 것으로 변형기반 학습방식은 일정량의 학습 데이터 이상이 되면 일정한 정확도를 유지하는 것을 보여준다.

5. 결론

본 논문에서는 코퍼스를 활용하여서 음운변환을 연구하였다. 이러한 작업을 위해서 음운변환 코퍼스를 활용하였으며, Brill(1992)의 변형기반 학습방법과 코퍼스에서 발견된 확률을 적용하였다. Brill(1992)의 변형기반의 학습방식은 음운변환이 아닌 형태 태깅작업에 활용하는 것이므로, 음운변환에 맞게 음운환경을 검사하는 문맥 윈도우를 활용하였다. 확률 적용은 발견되는 모든 환경에서 적용되는 규칙의 확률을 연산하여서 적용하였다.

본 연구는 음운변환의 작업에 코퍼스를 적용하였고, 학습기반의 방식을 활용하였다는 점에서 의의가 있다. 작업의 결과 Brill(1992)의 적용에서처럼 일정한 양의 코퍼스가 학습의 최대효과를 가져오는 것이 발견되었다. 또한 확률적용은 처리의 정확도의 향상을 가져왔다. 또한 전산음운론에서 코퍼스를 도입하는 것이 효과적이라는 점을 보였다.

코퍼스가 음운처리의 발전에 어떠한 영향을 미치고, 어떠한 형태의 코퍼스 작업이 효율적인지에 대한 부분은 앞으로의 연구로 남겨둔다. 그리고, 본 연구가 복잡한 음운규칙을 전통적인 '규칙순서화'를 통한 작업과 비교하여서 어떠한 장점이 있는지에 대한 연구도 매우 흥미로울 것이다. 특히, 음운규칙은 매우 복잡하며, '규칙순서화'는 기하급수적으로 증가된다. 따라서, 이러한 향후 연구는 매우 복잡한 연산을 다루어하므로, 앞으로의 이 연구의 발전적 가능성으로 남겨둔다. 또한 기존의 이은영 외(2005)에서 보여준 정확도에 비해서 현 연구의 정확도는 낮다. 이러한 점은 기계학습에 기반한 작업보다 정확한 데이터에 근거한 손으로 만든 규칙(hand-writing rules)에 기반한 작업이 더 정확하다는 것을 의미한다. 더나아가서 음운변환의 작업에서 실용성의 문제점을 기계학습과 손으로 만든 규칙기반으로 나누어서 생각해 볼 수 있다. 따라서, 본 연구의 정확도를 높이는 작업은 연구의 실용적 가능성을 고려하는 측면에서 의미가 있을 것이다.

참고 문헌

[1] 신지영, 차재은: 우리말 소리의 체계. 한국문화사 (2003).

[2] 이은영 외: 한국어 표준 발음 IPA 변환기. 한국인지과학회 춘계학술대회(2005).

[3] Albright A: Modeling analogy as probabilistic grammar. In Juliette Blevins, (ed.), Analogy in Grammar: Form and Acquisition. Oxford University Press(출간예정)

[3] Brill E.: Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. Computational

Linguistics (1995) 21(4).
[4] Daelemans W., S. Gillis and G. Durieux: The acquisition of stress: A data-oriented approach. Computational Linguistics (1994) 20(3).
[5] Gildea D. and D. Jurafsky: Learning bias and phonological-rule induction. Computational Linguistics (1996) 22(4).
[6] Hong L. and R. Huang: Design and implementation of AGTS Probabilistic Tagger. ICAME Journal(1998) 22.
[7] Jurafsky D. and D. Martin: Speech Processing and Natural Language Processing, Pritence Hall(2000).
[8] Kaplan R. and M. Kay: Regular models of phonological rule system. Computational Linguistics (1994). 20(3).
[9] Karttunen L.: Finite-state constraints, In Goldsmith, editor, The Last Phonological Rule, University of Chicago Press, Chicago (1993).
[10] Karttunen L.: The Proper Treatment of Optimality in Computational Phonology. FSMNLP (1998).
[11] Koskenniemi, K.: Two-level morphology. Department of General Linguistics, University of Helsinki (1983).